CSC535: Probabilistic Graphical Models

Variational Autoencoder

Prof. Jason Pacheco

Material adapted from:
https://lilianweng.github.io/posts/2018-08-12-vae/

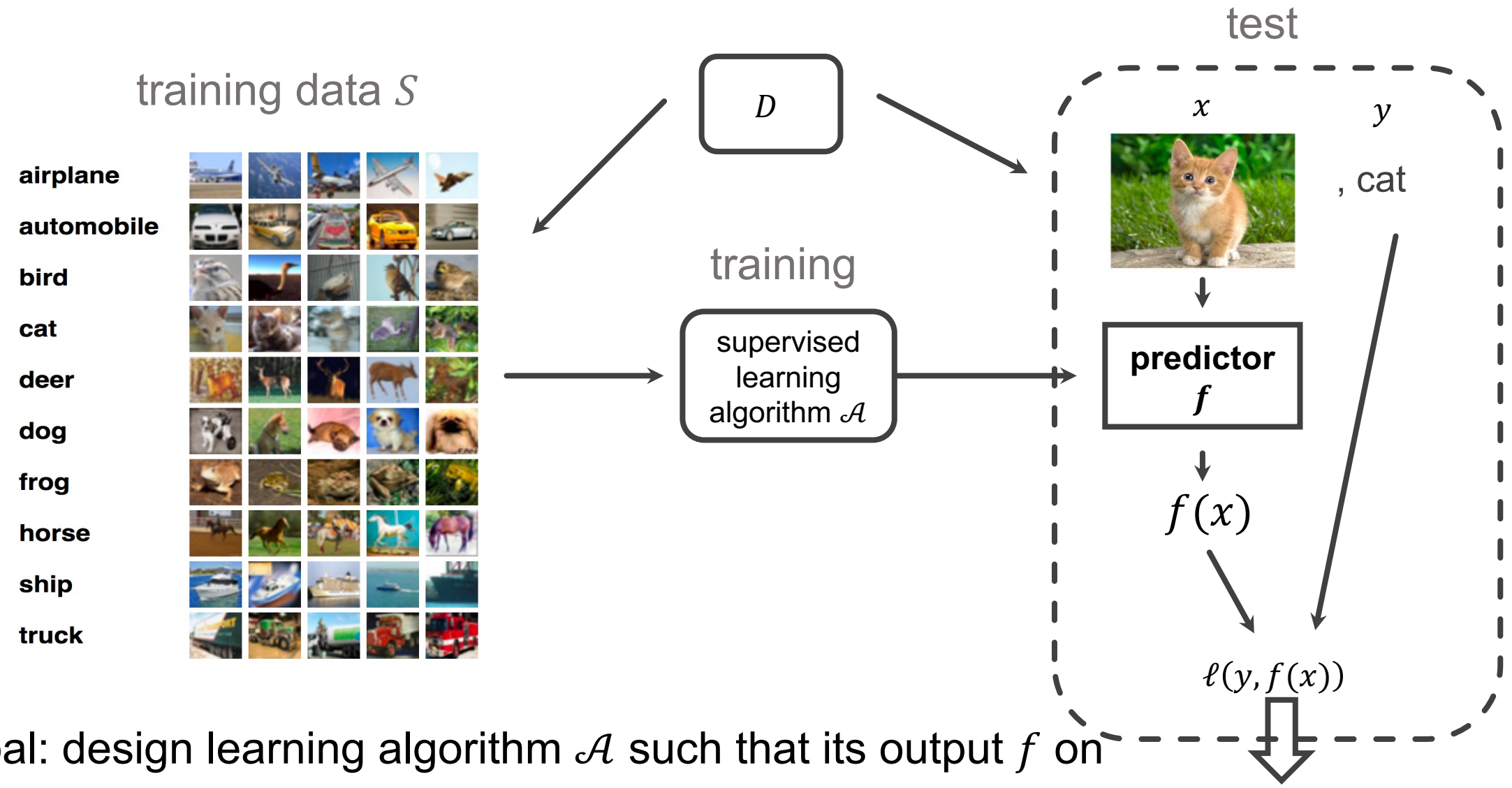
Outline

- Autoencoder
- Variational Autoencoder
- Beta-VAE

Outline

- Autoencoder
- Variational Autoencoder
- Beta-VAE

Supervised Learning

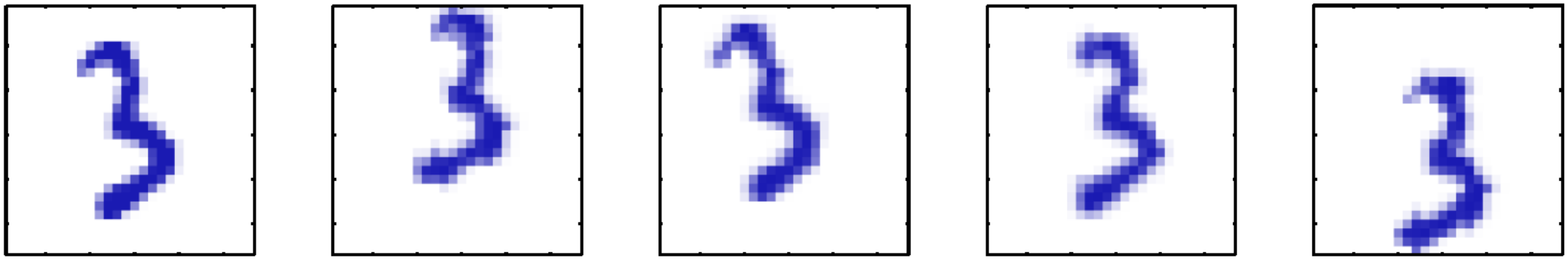


- Goal: design learning algorithm $\mathcal{A}$ such that its output f on iid training data S has low generalization error

Generalization error: $L_D(f) = \mathrm{E}_{(x,y)\sim D}\, \ell(y, f(x))$

Dimensionality Reduction

Data often have a lot of redundant information…



Example A dataset consisting of a hand-drawn 3 at random locations and rotations in a 100x100 pixel image.

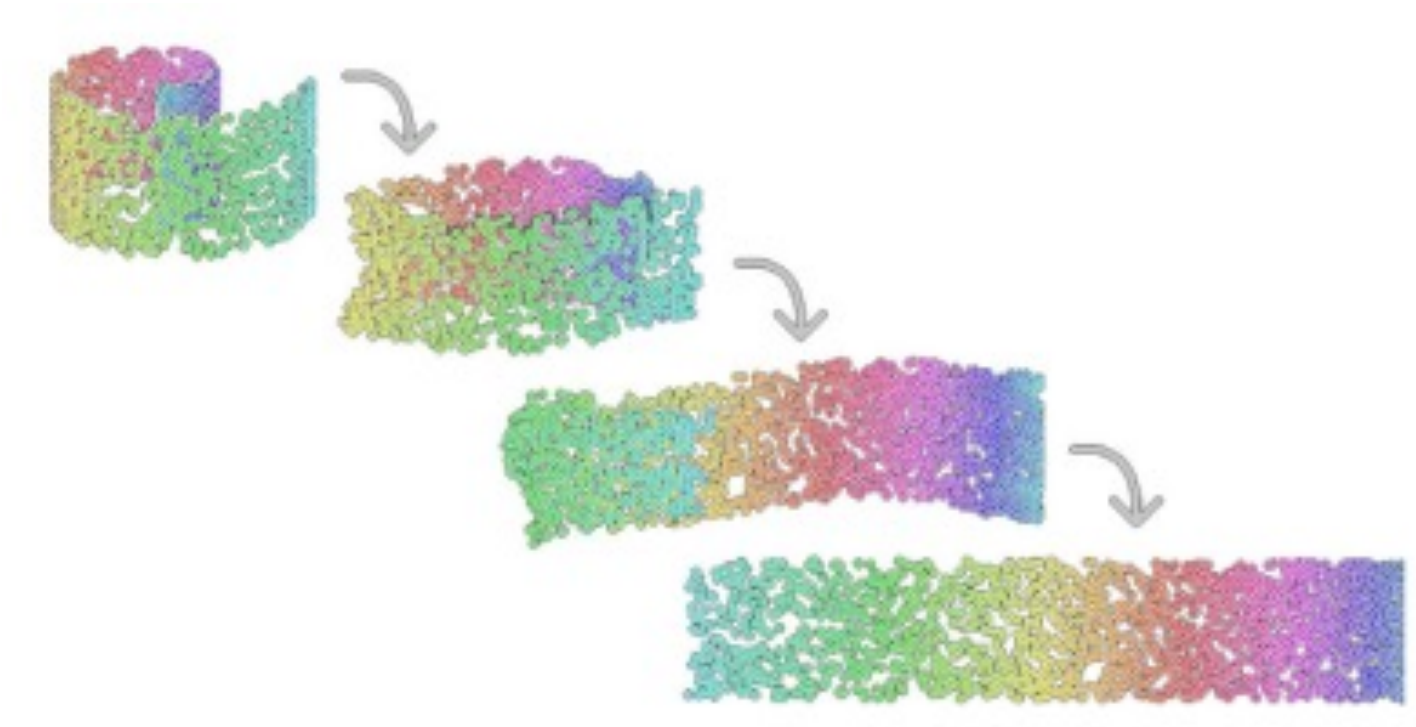
Data Dimension 100 x 100 = 10,000

Intrinsic Dimension 3 (X-position, Y-position, Rotation)

[Source: Bishop, C.]

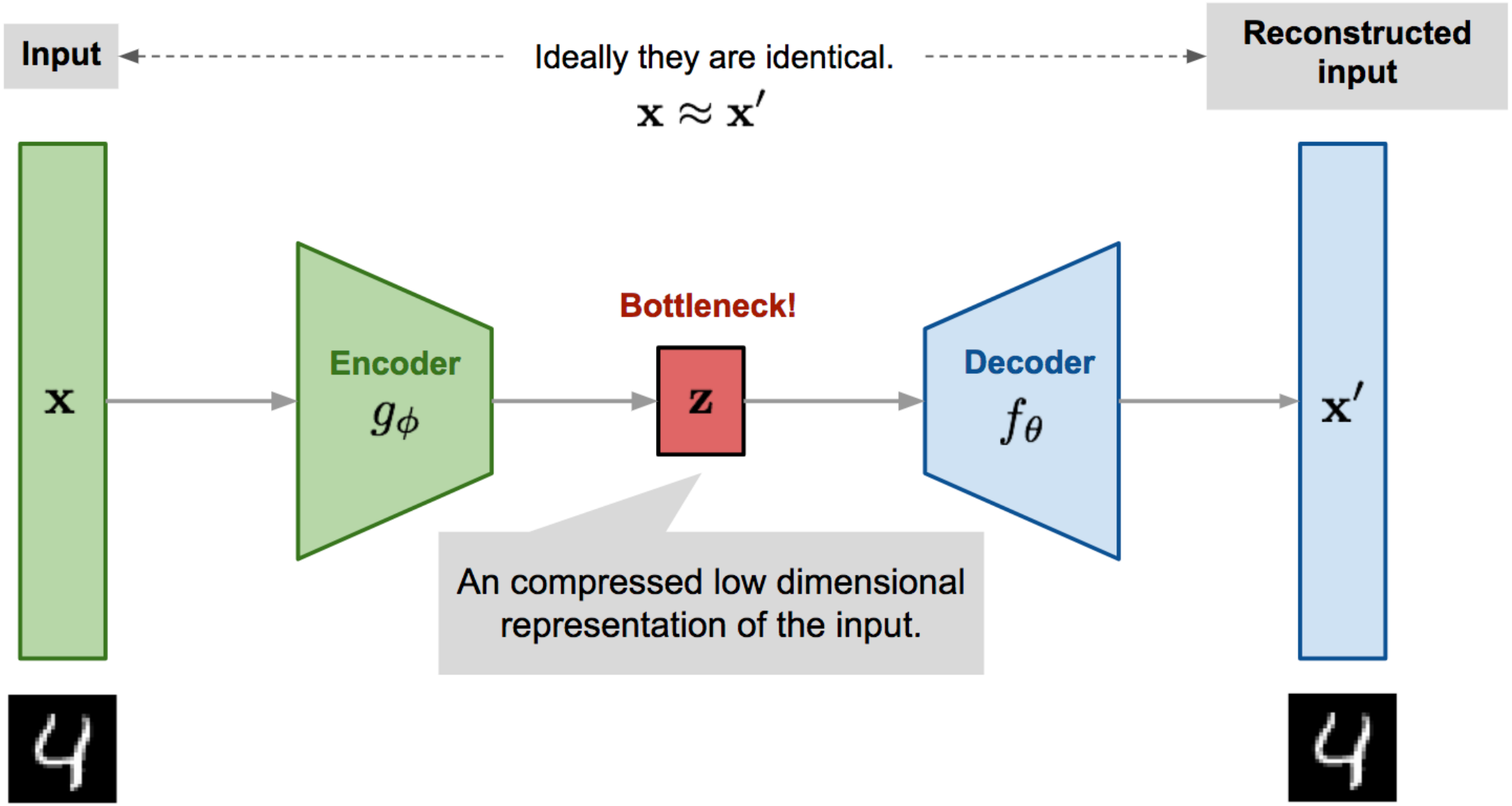
Dimensionality Reduction : Manifold Hypothesis

…or data are high-dimensional and hard to visualize…



…in all cases finding lower *intrinsic dimension* is useful

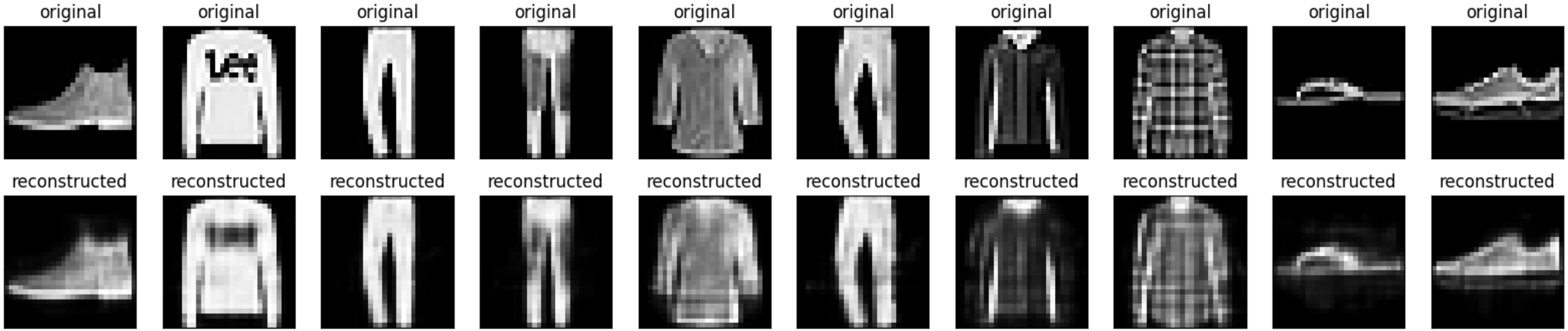
Autoencoder



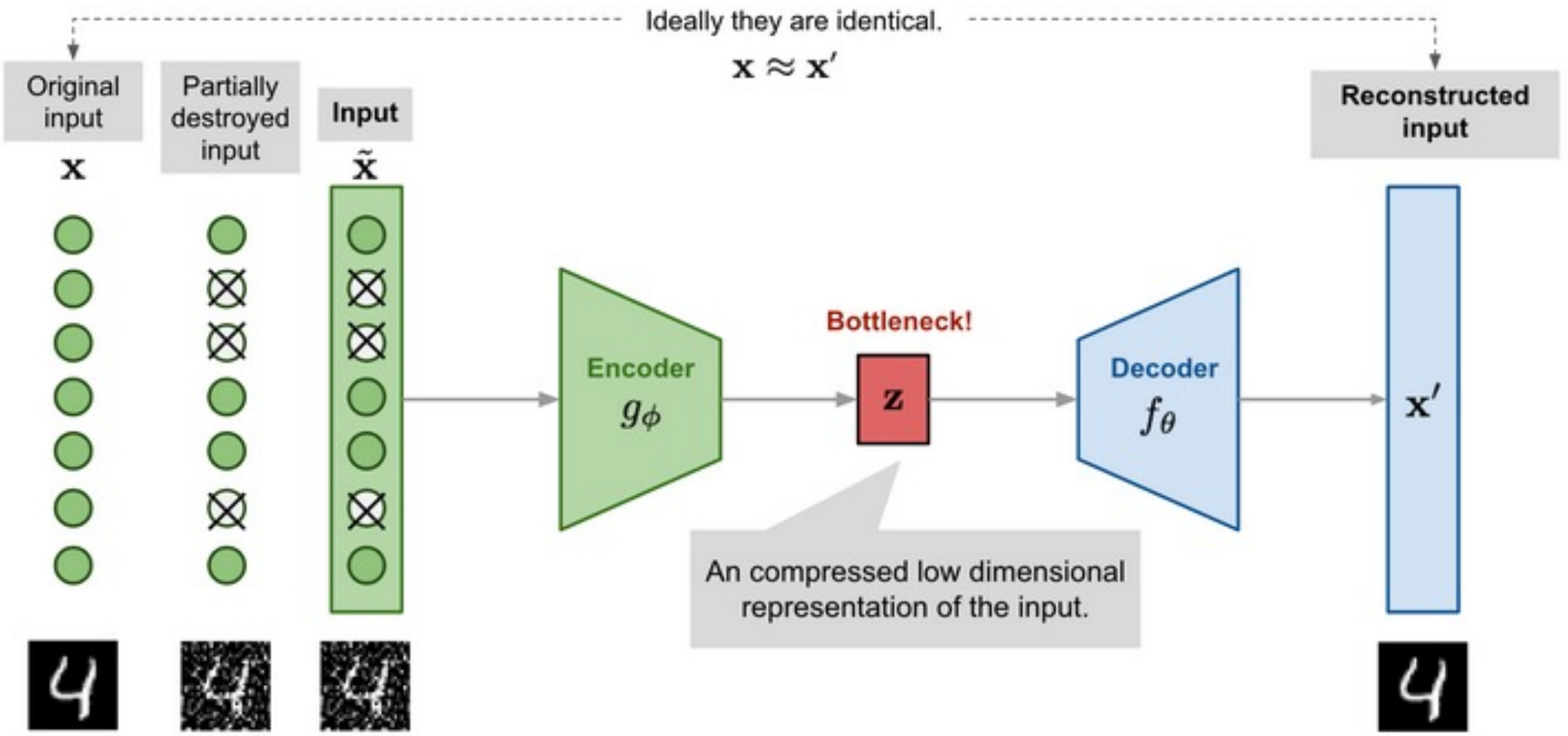
Learning

Mean squared error (MSE) reconstruction loss:

$$L_{\mathrm{AE}}(\theta, \phi) = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{x}^{(i)} - f_\theta(g_\phi(\mathbf{x}^{(i)})))^2$$



Denoising Autoencoder



Denoising Autoencoder

$$\tilde{\mathbf{x}}^{(i)} \sim \mathcal{M}_{\mathcal{D}}(\tilde{\mathbf{x}}^{(i)}|\mathbf{x}^{(i)})$$

$$L_{\mathrm{DAE}}(\theta, \phi) = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}^{(i)} - f_\theta(g_\phi(\tilde{\mathbf{x}}^{(i)})))^2$$

- Humans can easily recognize a scene even when some inputs are corrupted
- Conceptually, we “repair” the input in our brains
- For high-dimensional input the model depends on evidence from many input dims
- Prevents overfitting to any single data dimension (more robust)
- Noise is controlled by $\mathcal{M}_{\mathcal{D}}(\tilde{\mathbf{x}}^{(i)}|\mathbf{x}^{(i)})$ and can be adapted to any noise model

Sparse Autoencoder

- Common activation functions: sigmoid, tanh, relu, leaky relu, etc.
- Neuron is *activated* when activation function is near 1, and *inactive* otherwise
- Sparse Autoencoder encourages model to have a small number of neurons active
- Avoids overfitting, leads to more robust learning

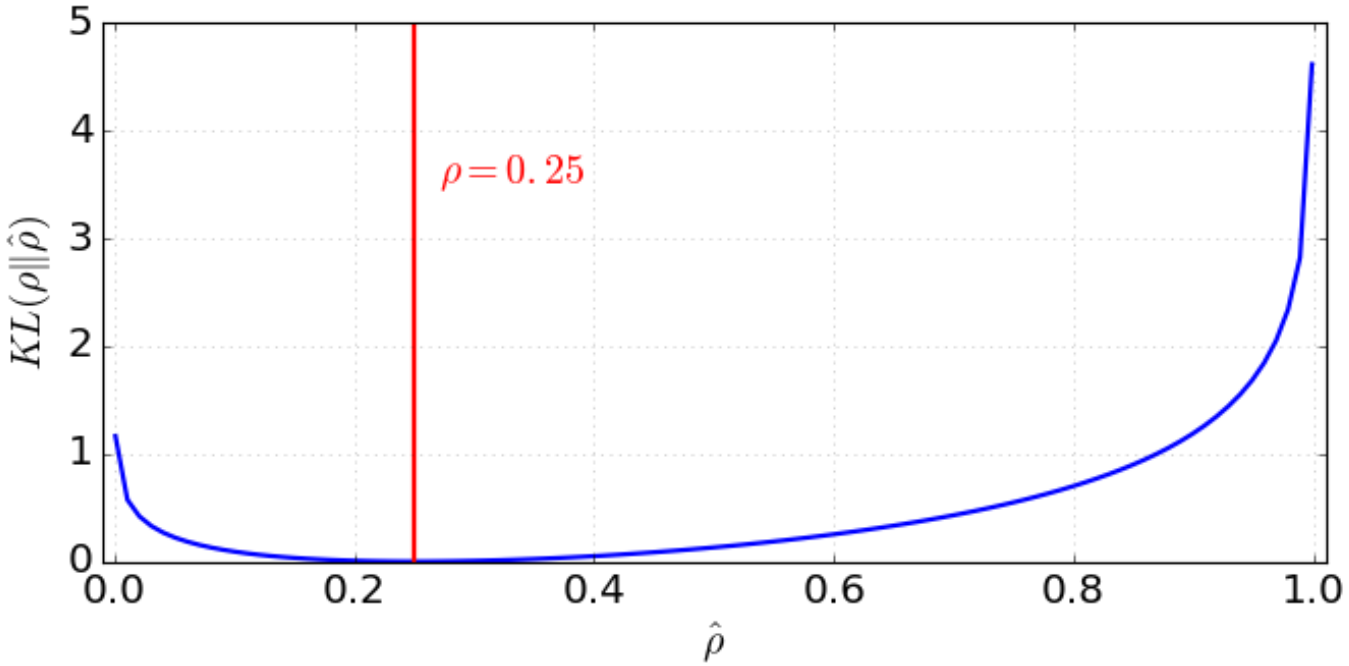
- s_l : Neurons in *l*-th hidden layer
- $a_j^{(l)}$: Activation function for *j*-th neuron in *l*-th layer
- Fraction of active neurons expected to be small (e.g. $\rho = 0.05$)

$$\hat{\rho}_j^{(l)} = \frac{1}{n} \sum_{i=1}^{n} [a_j^{(l)}(\mathbf{x}^{(i)})] \approx \rho$$

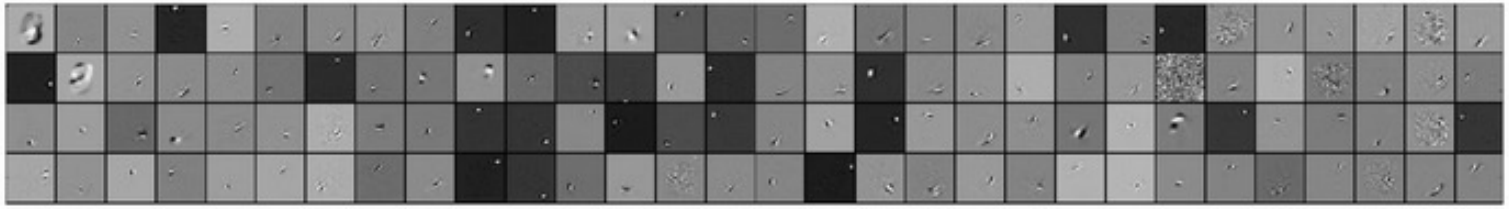
Sparse Autoencoder

Achieve sparsity constraint by adding penalty to loss function,

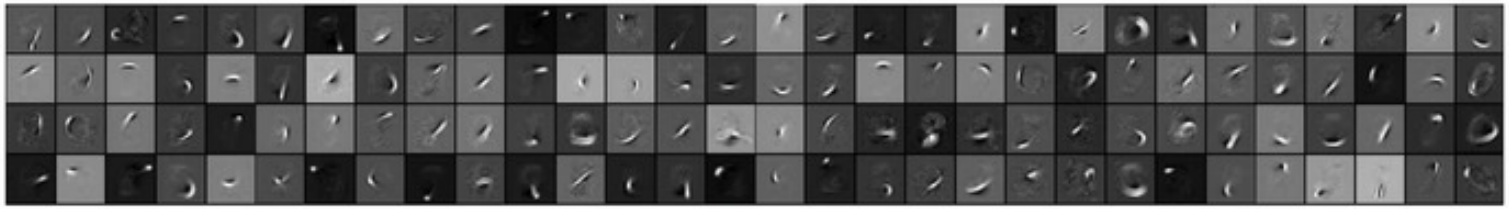
$$L_{\mathrm{SAE}}(\theta) = L(\theta) + \beta \sum_{l=1}^{L} \sum_{j=1}^{s_l} D_{\mathrm{KL}}(\rho \| \hat{\rho}_j^{(l)})$$



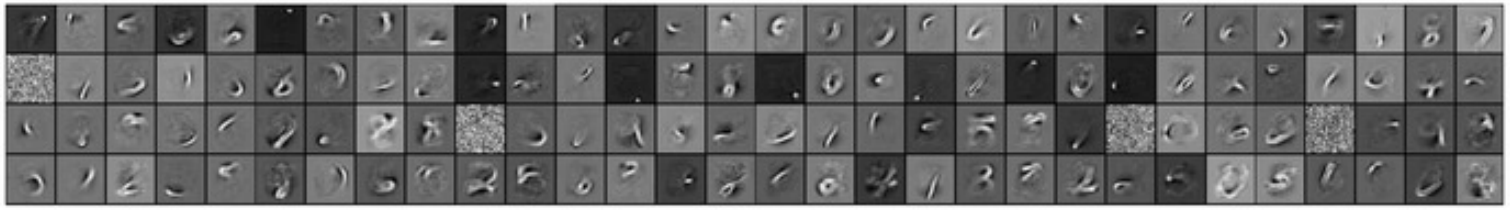
k-Sparse Autoencoder



(a) $k = 70$



(b) $k = 40$



(c) $k = 25$

(d) $k = 10$

Fig. 5. Filters of the k-sparse autoencoder for different sparsity levels k, learnt from MNIST with 1000 hidden units.. (Image source: Makhzani and Frey, 2013)

Outline

- Autoencoder
- Variational Autoencoder
- Beta-VAE

Variational Autoencoder

- Autoencoder
  - Learns single encoding
  - Outputs single reconstruction
  - Can be brittle

- Variational Autoencoder (VAE)
  - Learn distribution over encoding and data—more robust learning
  - Prior: $p_\theta(z)$
  - Likelihood (decoder): $p_\theta(x \mid z)$
  - Posterior (encoder): $p_\theta(z \mid x)$

VAE Generative Process

Assuming we know the real parameters θ^* generate a new data point $x^{(i)}$:

1. First, sample a $z^{(i)}$ from the prior distribution $p_{\theta^*}(z)$
2. Then generate data $x^{(i)}$ from the *decoder* $p_{\theta^*}(x \mid z^{(i)})$

Optimal parameter is the one that maximizes probability of the data:

$$\theta^* = \arg\max_{\theta} \prod_{i=1}^{n} p_\theta(\mathbf{x}^{(i)})$$

Or equivalently the maximum log-likelihood:

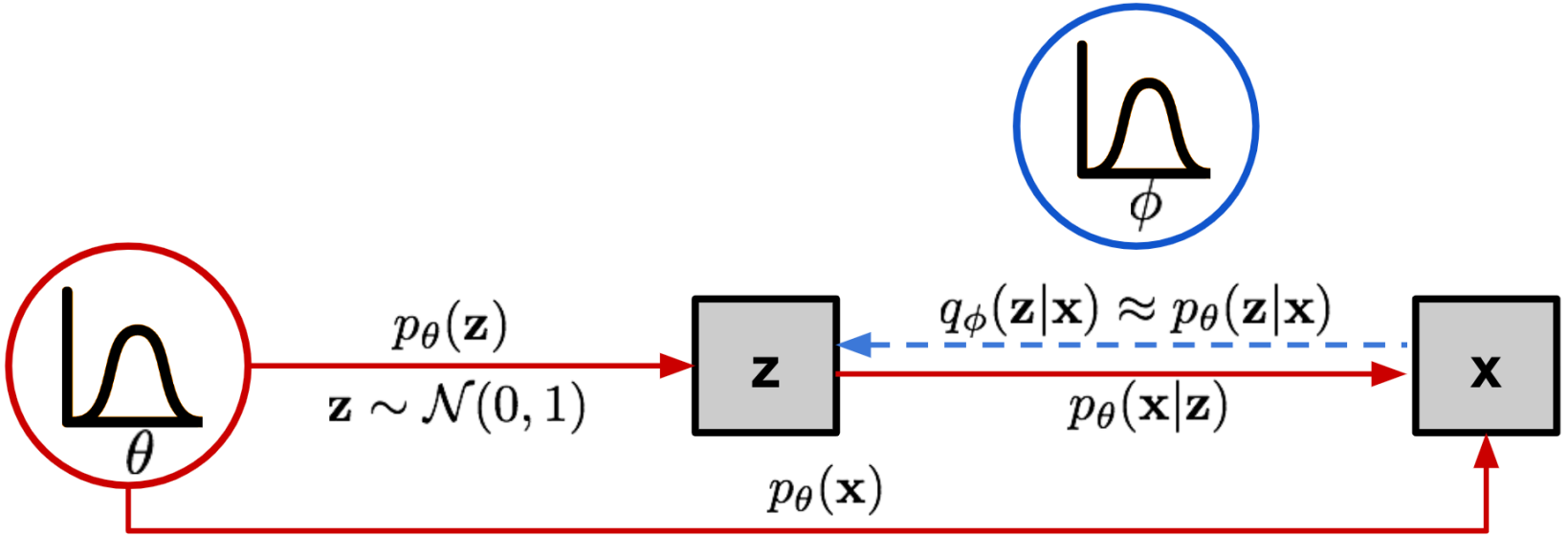
$$\theta^* = \arg\max_{\theta} \sum_{i=1}^{n} \log p_\theta(\mathbf{x}^{(i)})$$

VAE Learning

Marginal likelihood given by,

$$p_\theta(\mathbf{x}^{(i)}) = \int p_\theta(\mathbf{x}^{(i)}|\mathbf{z})p_\theta(\mathbf{z})d\mathbf{z}$$

Typically lacks a closed-form solution...



VAE Inference

$$D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z}|\mathbf{x}))$$
$$= \int q_\phi(\mathbf{z}|\mathbf{x}) \log \frac{q_\phi(\mathbf{z}|\mathbf{x})}{p_\theta(\mathbf{z}|\mathbf{x})} d\mathbf{z}$$

VAE Inference

$$
\begin{aligned}
& D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z}|\mathbf{x})) \\
&= \int q_\phi(\mathbf{z}|\mathbf{x}) \log \frac{q_\phi(\mathbf{z}|\mathbf{x})}{p_\theta(\mathbf{z}|\mathbf{x})} d\mathbf{z} \\
&= \int q_\phi(\mathbf{z}|\mathbf{x}) \log \frac{q_\phi(\mathbf{z}|\mathbf{x}) p_\theta(\mathbf{x})}{p_\theta(\mathbf{z},\mathbf{x})} d\mathbf{z} && ;\ \text{Because } p(z|x)=p(z,x)/p(x) \\
&= \int q_\phi(\mathbf{z}|\mathbf{x}) \big(\log p_\theta(\mathbf{x}) + \log \frac{q_\phi(\mathbf{z}|\mathbf{x})}{p_\theta(\mathbf{z},\mathbf{x})} \big) d\mathbf{z} \\
&= \log p_\theta(\mathbf{x}) + \int q_\phi(\mathbf{z}|\mathbf{x}) \log \frac{q_\phi(\mathbf{z}|\mathbf{x})}{p_\theta(\mathbf{z},\mathbf{x})} d\mathbf{z} && ;\ \text{Because } \int q(z|x)dz=1 \\
&= \log p_\theta(\mathbf{x}) + \int q_\phi(\mathbf{z}|\mathbf{x}) \log \frac{q_\phi(\mathbf{z}|\mathbf{x})}{p_\theta(\mathbf{x}|\mathbf{z}) p_\theta(\mathbf{z})} d\mathbf{z} && ;\ \text{Because } p(z,x)=p(x|z)p(z) \\
&= \log p_\theta(\mathbf{x}) + \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})} [\log \frac{q_\phi(\mathbf{z}|\mathbf{x})}{p_\theta(\mathbf{z})} - \log p_\theta(\mathbf{x}|\mathbf{z})] \\
&= \log p_\theta(\mathbf{x}) + D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z})) - \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{z})
\end{aligned}
$$

VAE Inference

So we have:

$$D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z}|\mathbf{x})) = \log p_\theta(\mathbf{x}) + D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z})) - \mathbb{E}_{q_\phi} \log p_\theta(\mathbf{x}|\mathbf{z})$$

Rearranging terms we have:

$$\underbrace{\log p_\theta(\mathbf{x})}_{\text{Marginal Likelihood}} - \underbrace{D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z}|\mathbf{x}))}_{\text{Bound Gap}} = \underbrace{\mathbb{E}_{q_\phi} \log p_\theta(\mathbf{x}|\mathbf{z}) - D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z}))}_{\text{Evidence Lower Bound (ELBO)}}$$

Formulate as minimizing loss function:

$$\theta^*, \phi^* = \arg\min_{\theta,\phi} -\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{z}) + D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z}))$$

Reparameterization Trick

$$\theta^*, \phi^* = \arg\min_{\theta,\phi} -\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{z}) + D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \| p_\theta(\mathbf{z}))$$

No straightforward Monte Carlo estimator of gradient…

$$\begin{aligned}\frac{\partial}{\partial\phi}\mathbb{E}_{q_\phi}[f(z,\phi)] &= \int \frac{\partial}{\partial\phi} q_\phi(z \mid x) f(z,\phi)\, dz \\ &= \int q'_\phi(z \mid x) f(z,\phi)\, dz + \int q_\phi(z \mid x) f'(z,\phi)\, dz\end{aligned}$$

…need to use *reparameterization trick*.

Gaussian Reparameterization

So we need a deterministic function s.t. $\mathbf{z} = g(\phi, x, \epsilon)$

Suppose we want to sample a Gaussian RV,

$$\mathbf{z} \sim \mathcal{N}(\mu(x), \sigma^2(x))$$

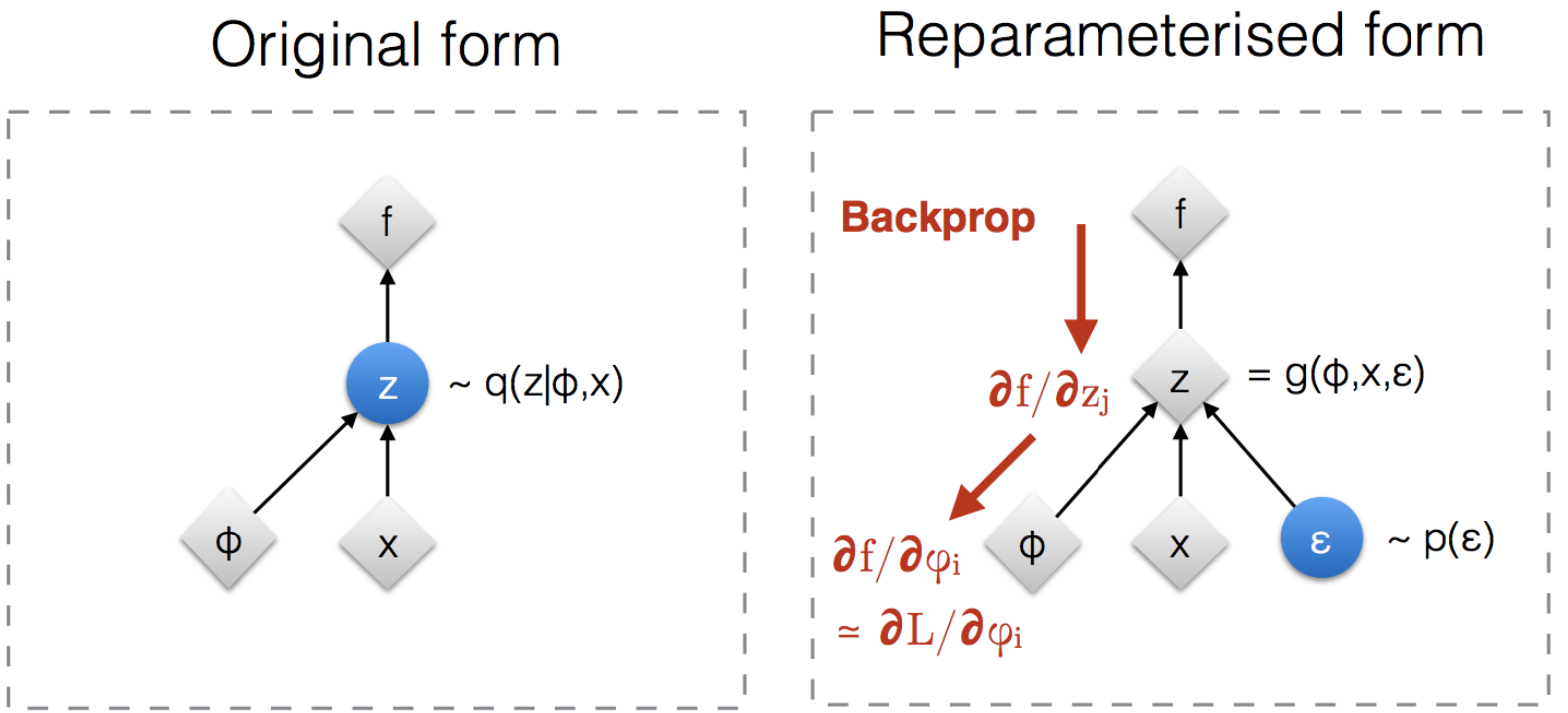
But we only know how to sample a *standard* Gaussian RV,

$$\epsilon \sim \mathcal{N}(0, 1)$$

Gaussians are closed under linear transformations so,

$$\mathbf{z} = \mu(x) + \underbrace{\sigma(x)\epsilon}_{\mathbf{z} = g(\phi, x, \epsilon)} \sim \mathcal{N}(\mu, \sigma^2)$$

Reparameterization Trick

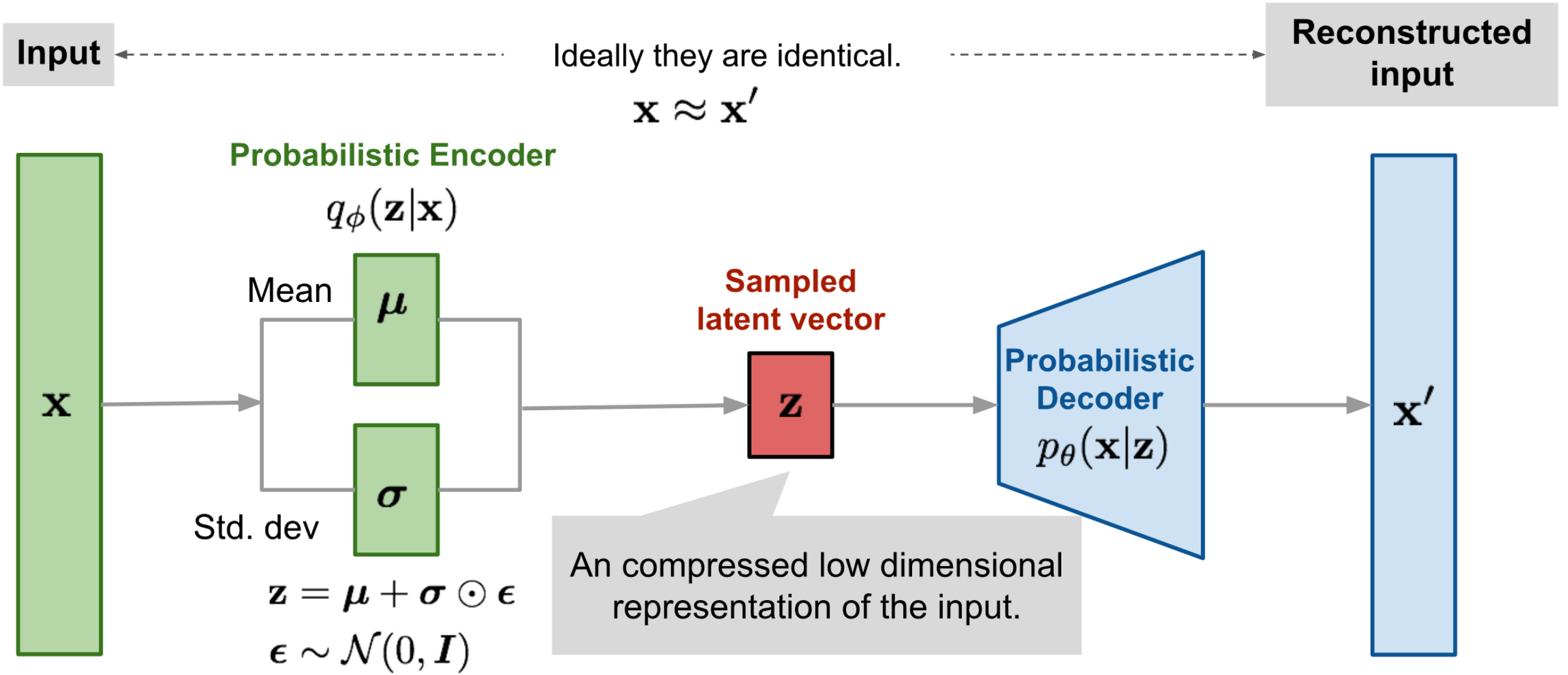


: Deterministic node

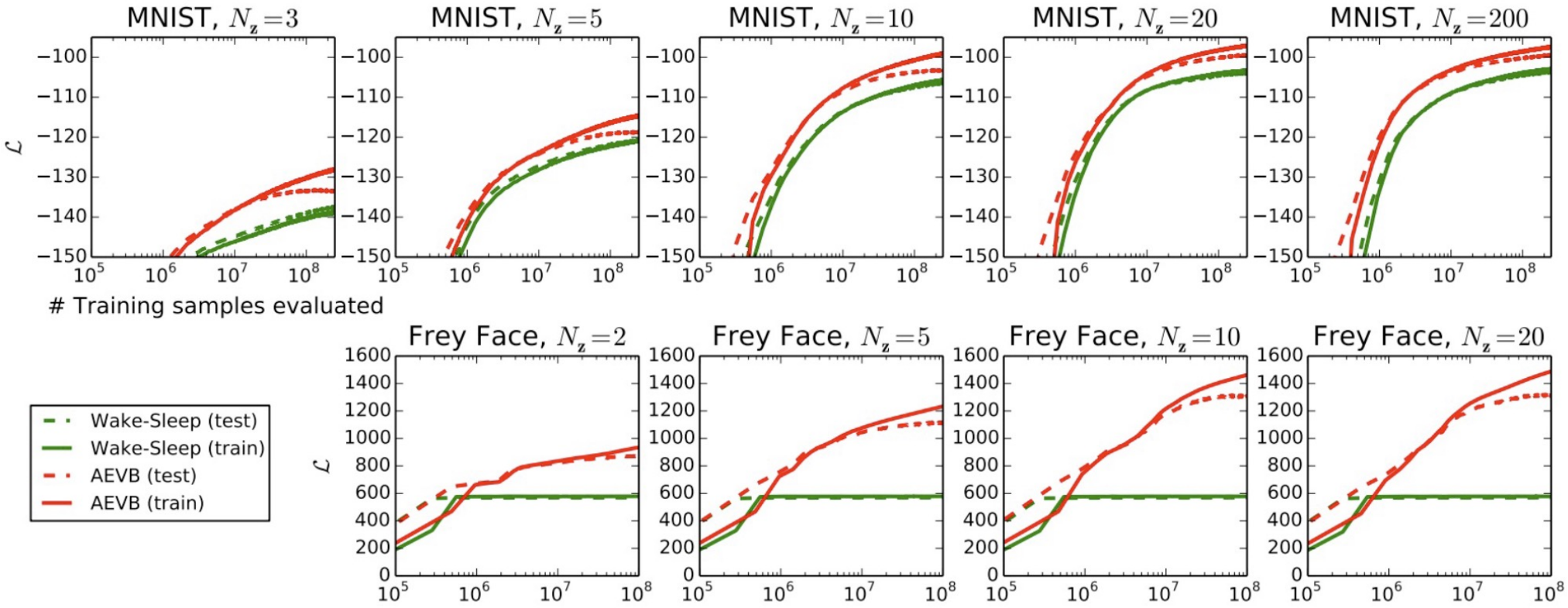
: Random node

[Kingma, 2013]
[Bengio, 2013]
[Kingma and Welling 2014]
[Rezende et al 2014]

Variational Autoencoder

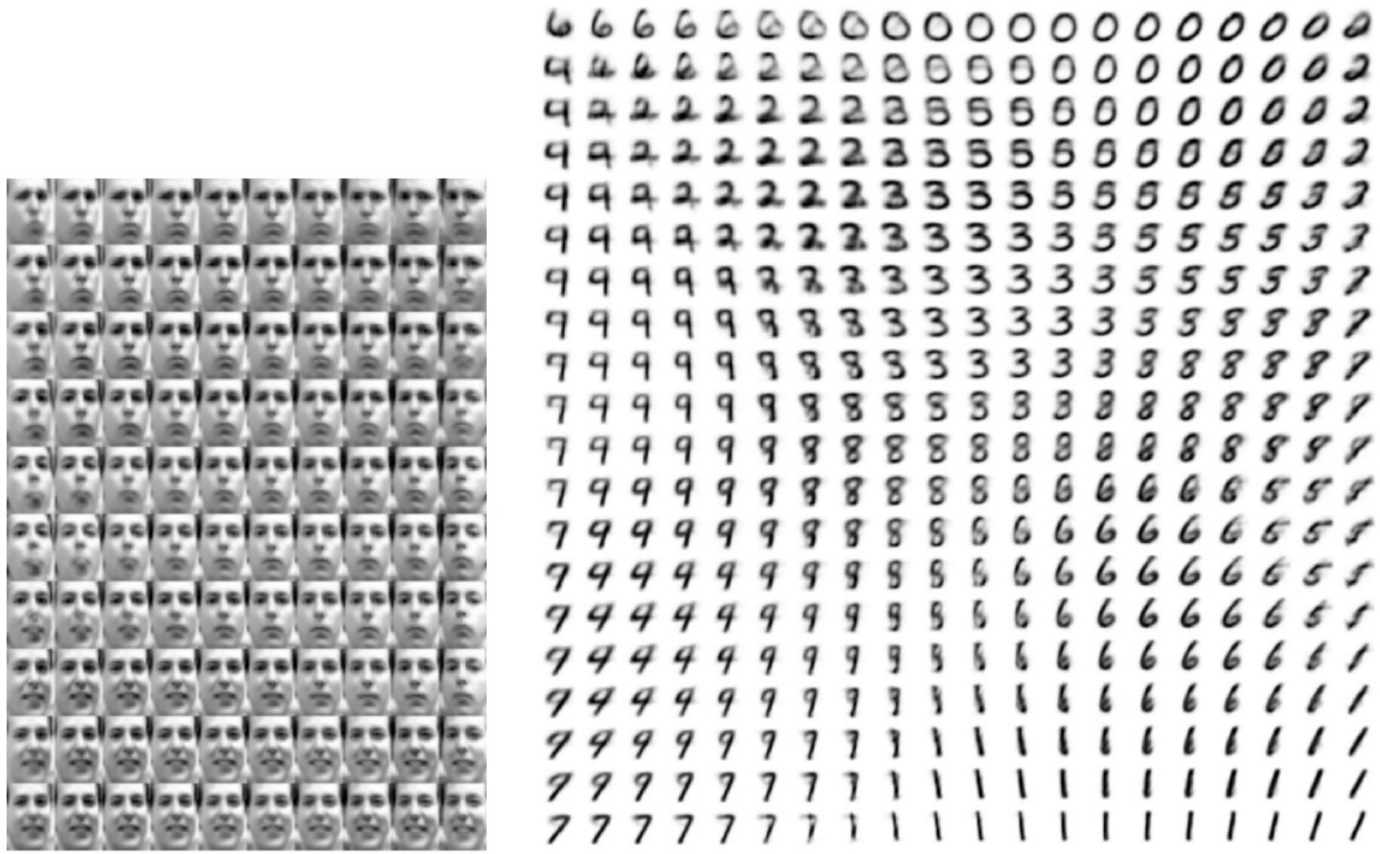


MNIST Likelihood Lower Bound



Vertical axis: the estimated average variational lower bound per datapoint. The estimator variance was small (< 1) and omitted. **Horizontal ax**is: amount of training points evaluated. N_z : dim. of latent space

Visualization of High-Dimensional Data



(a) Learned Frey Face manifold

(b) Learned MNIST manifold

Figure 4: Visualisations of learned data manifold for generative models with two-dimensional latent space, learned with AEVB. Since the prior of the latent space is Gaussian, linearly spaced coordinates on the unit square were transformed through the inverse CDF of the Gaussian to produce values of the latent variables $\mathbf{z}$. For each of these values $\mathbf{z}$, we plotted the corresponding generative $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ with the learned parameters $\boldsymbol{\theta}$.

Visualization of High-Dimensional Data

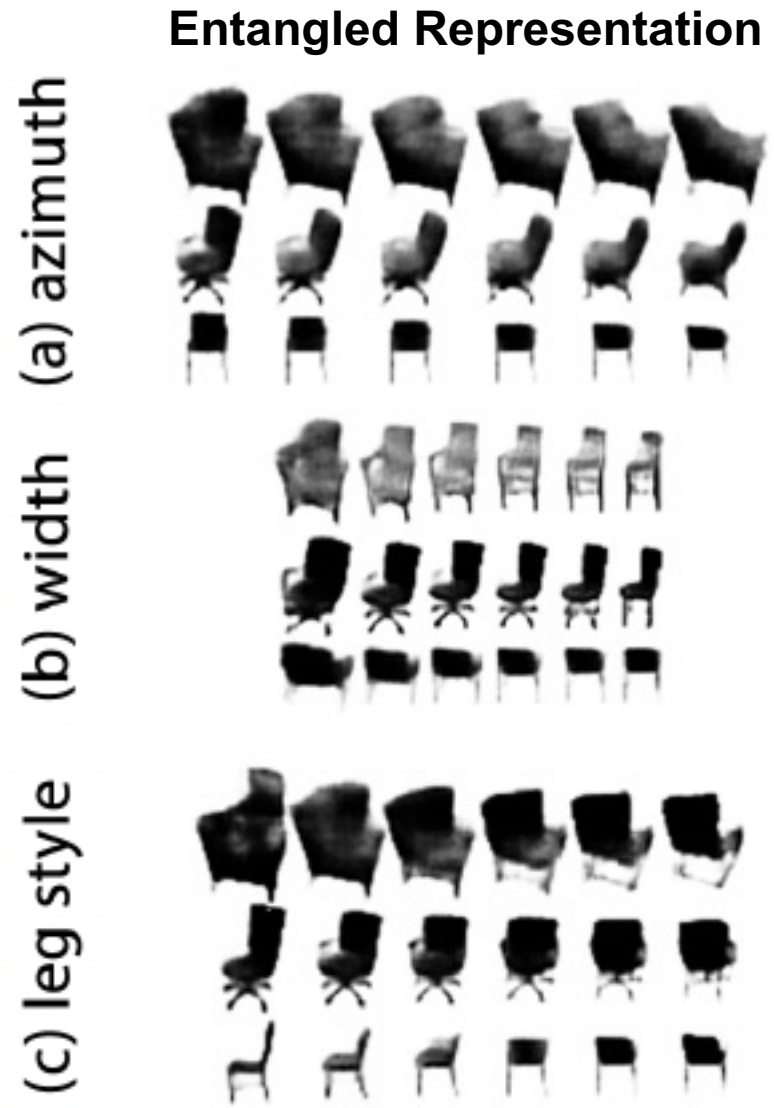
(a) 2-D latent space (b) 5-D latent space (c) 10-D latent space (d) 20-D latent space

Figure 5: Random samples from learned generative models of MNIST for different dimensionalities of latent space.

Outline

- Autoencoder
- Variational Autoencoder
- Beta-VAE

Entanglement



Code **z** learned by VAE is fully correlated in the posterior…

…this leads to a behavior known as *entanglement…*

…for interpretable codes we prefer them to control independent aspects of data generation, known as a *disentangled representation*

Beta-VAE

Consider the constrained optimization problem:

$$\max_{\phi,\theta} \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}[\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{z})]$$

$$\text{subject to } \underbrace{D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z})) < \delta}$$

Constrains posterior (encoder) to be closer to prior

Independent Gaussian prior,

$$p_\theta(\mathbf{z}) = \mathcal{N}(0, I)$$

encourages independent codes in the posterior (disentanglement)

Beta-VAE

Formulate the Lagrangian as,

$$\begin{aligned}
\mathcal{F}(\theta, \phi, \beta) &= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{z}) - \beta(D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z})) - \delta) \\
&= \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{z}) - \beta D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z})) + \beta\delta \\
&\geq \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{z}) - \beta D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z})) \qquad ; \text{Because } \beta,\delta \geq 0
\end{aligned}$$

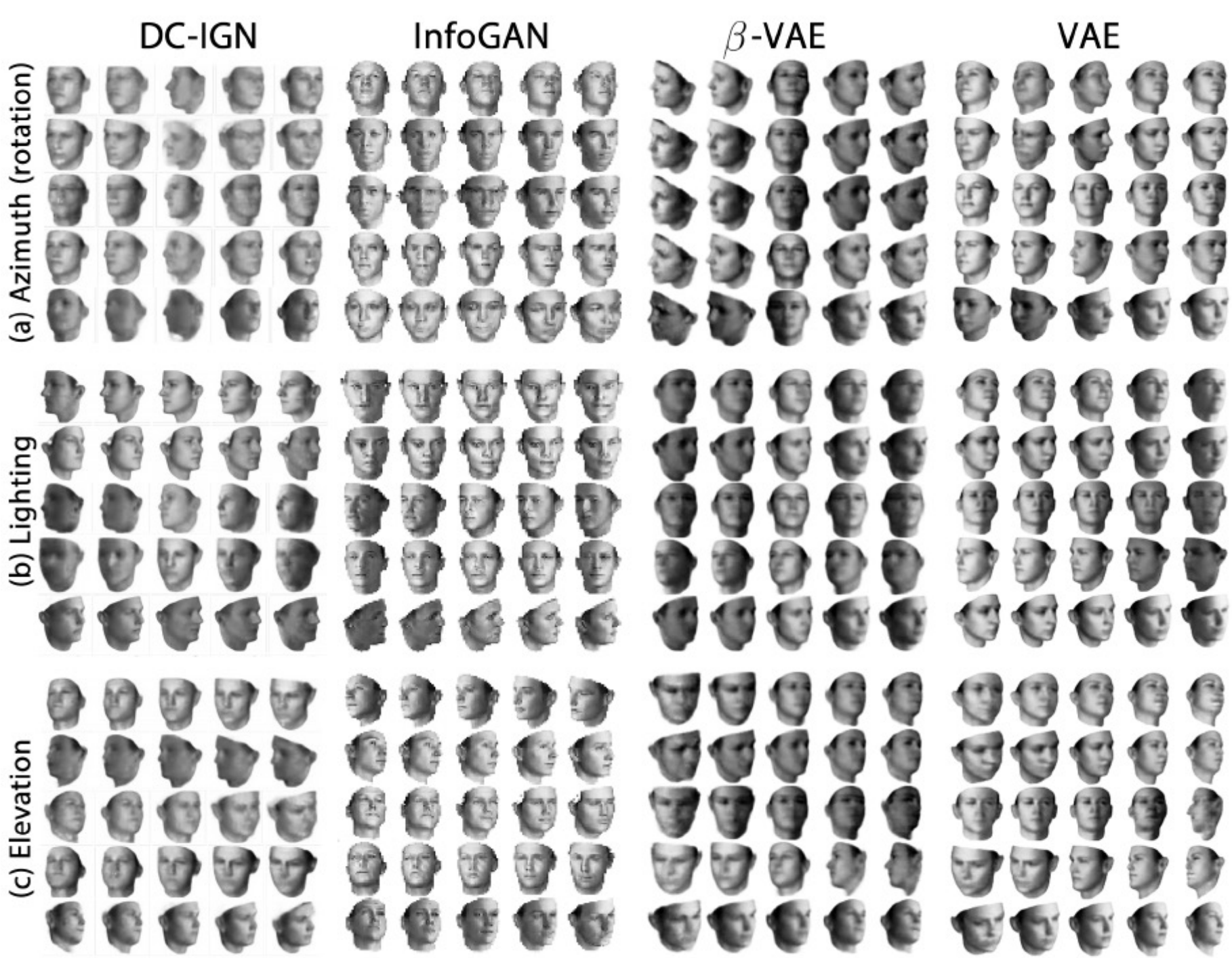
So we have the Beta-VAE loss:

$$L_{\mathrm{BETA}}(\phi, \beta) = -\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{z}) + \beta D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z}))$$

Controls degree of disentanglement

Identical to VAE loss, but with additional control on disentaglement


β-VAE
VAE
(a) Azimuth (rotation)
(b) emotion (smile)
(c) hair (fringe)


DC-IGN
InfoGAN
β-VAE
VAE
(a) Azimuth (rotation)
(b) Lighting
(c) Elevation

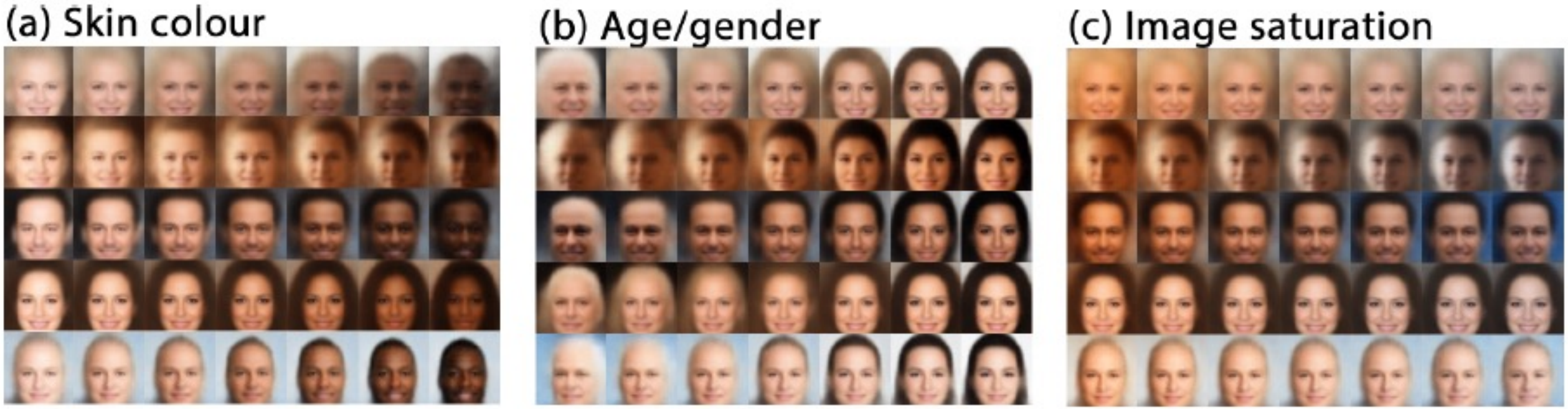


Figure 4: **Latent factors learnt by β-VAE on celebA:** traversal of individual latents demonstrates that β-VAE discovered in an unsupervised manner factors that encode skin colour, transition from an elderly male to younger female, and image saturation.